



مقدمه

در این تمرین، با بکارگیری مفاهیمی که در طول درس فراگرفته‌اید، سیستمی مشابه دراپ‌باکس را پیاده‌سازی خواهید کرد. این تمرین در دو فاز ارائه می‌شود. در فاز اول، هسته‌ی برنامه و رابط کاربری خط فرمان^۱ آن را پیاده‌سازی می‌کنید و در فاز دوم، واسط گرافیکی سیستم را بر مبنای فریم‌ورک‌های وب پیاده‌سازی کرده و سیستم خود را به یک سرور برای بارگیری و بارگذاری فایل‌ها متصل خواهید کرد.

ای‌پی درایو^۲

ای‌پی درایو یک ارائه‌کننده‌ی خدمات میزبانی فایل^۳ در وب است که امکان همگام‌سازی^۴ فایل‌ها روی چند رایانه‌ی مختلف یا بین چند کاربر مختلف، از طریق اینترنت را فراهم می‌کند. کارکرد برنامه‌ای که شما پیاده‌سازی خواهید کرد شبیه فایل سیستم^۵ سیستم عامل لینوکس است که امکان بارگذاری^۶ و بارگیری^۷ فایل‌ها بر روی شبکه را نیز خواهد داشت. در برنامه‌ی شما کاربران ابتدا در سیستم ثبت‌نام کرده، سپس می‌توانند به حساب کاربری خود وارد شوند. پس از آن کاربران با اجرای دستورات مختلف می‌توانند با فایل‌های موجود در برنامه کار کنند. سیستمی که شما پیاده‌سازی می‌کنید در واقع یک فایل سیستم مجازی است.

اجزای سیستم

فایل^۸ و پوشه^۹

عناصری که کاربران با آن در این برنامه کار خواهند کرد، فایل‌ها و پوشه‌ها هستند. هر پوشه می‌تواند شامل تعدادی فایل یا پوشه‌ی دیگر باشد. شیوه‌ی سازماندهی فایل‌ها و پوشه‌های مربوط به هر کاربر به شیوه‌ی زیر است:

هر عنصر، یک مسیر در سیستم دارد که مشخص‌کننده جایگاه فعلی آن در سیستم است. مسیر یک عنصر، رشته‌ای است که شامل نام پوشه‌های آن مسیر و نام خود آن عنصر است که با / در کنار هم چاپ شده‌اند. مثلاً `amir/sample.txt` /مسیر یک فایل متنی در

^۱ Command Line

^۲ AP Drive

^۳ File Hosting Service

^۴ Synchronization

^۵ File System

^۶ Upload

^۷ Download

^۸ File

^۹ Directory or Folder

سیستم است. مسیر ریشه¹⁰ در سیستم، / است. به ازای هر کاربر یک پوشه با نام او در مسیر /userName ایجاد می شود که پوشه ی خانه¹¹ کاربر پس از ورود به سیستم، همین پوشه است. مالک¹² این پوشه، کاربر است و مجوزهای دسترسی خواندن و نوشتن را در این پوشه دارد.

یک مسیر می تواند مطلق¹³ یا نسبی¹⁴ باشد که منظور از مسیر مطلق، مسیر عنصر مورد نظر از ریشه بوده و مسیر نسبی، مسیر عنصر از پوشه فعلی می باشد. دقت کنید که هر مسیر مطلق با کاراکتر / آغاز می شود.

هر **عنصر** (فایل یا پوشه) ایجاد کننده ای دارد که به عنوان مالک آن در نظر گرفته می شود. همچنین هنگام ایجاد یک فایل، گروه مالک فایل، گروهی در نظر گرفته می شود که به اسم کاربر است. هر عنصر دقیقاً یک کاربر مالک و حداکثر یک گروه مالک دارد.

برای هر عنصر دو دسترسی خواندن و نوشتن تعریف می شود. مشاهده ی محتویات فایل و نسخه برداری¹⁵ از فایل نیاز به مجوز خواندن دارد. همچنین حذف کردن و جابه جایی فایل نیاز به مجوز نوشتن دارد. به صورت پیش فرض، تنها مالک مجوز خواندن و نوشتن یک عنصر را دارد. همچنین تنها کاربر مالک می تواند مجوزهای دسترسی¹⁶ به عنصر را تغییر دهد. در صورت عدم وجود مجوز، دسترسی به آن عنصر مجاز نمی باشد. به ازای هر جفت کاربر و عنصر، مجوزهای خاص می توانند وجود داشته باشند.

کاربر

در این سیستم، به جز دستور ورود، هر دستور توسط یک کاربر که از قبل به سیستم اضافه شده است اجرا می شود. هر کاربر، یکی از سه نقش زیر را می تواند داشته باشد که دارای سطح های دسترسی متفاوتی هستند:

- کاربر ممتاز¹⁷: کاربر ممتاز، بالاترین سطح دسترسی را در سیستم دارد.
- مدیر¹⁸ سیستم: مدیر سیستم می تواند دستورات مدیریتی را در سیستم اجرا کند.
- کاربر عادی: کاربر عادی کمترین سطح دسترسی را در سیستم دارد. به صورت پیش فرض در هنگام ثبت نام، این سطح دسترسی برای کاربر تعریف می شود. در طول برنامه امکان تغییر نقش این نوع کاربران توسط کاربر ممتاز امکان پذیر است.

گروه

یک گروه، مجموعه ای از کاربران را شامل می شود. برای این مجموعه می توان سطح دسترسی خاصی را به ازای هر عنصر تعیین کرد. نام گروه باید یکتا باشد و هر کاربر، به صورت پیش فرض، عضو گروهی با اسم نام کاربری خود می باشد. برای مثال، با ثبت نام یک کاربر با نام کاربری amir، یک گروه با اسم amir ایجاد می شود که این کاربر عضو آن است.

¹⁰ Root

¹¹ Home Directory

¹² Owner

¹³ Absolute Path

¹⁴ Relative Path

¹⁵ Copy

¹⁶ Permission Access

¹⁷ Superuser

¹⁸ Admin

دستورات

در ادامه دستوراتی که باید در سیستم پیاده‌سازی شوند شرح داده می‌شوند. به ازای هر دستور ابتدا توضیحی در رابطه با عملکرد دستور داده شده، سپس فرمت دستور تشریح شده است. در مورد فرمت دستورها توجه کنید که آرگومان‌هایی که داخل [] (براکت) قرار می‌گیرند اختیاری بوده و آرگومان‌هایی که داخل {} (آکولاد) می‌باشند اجباری هستند. برای اجرای همه‌ی دستورها به جز دستور ورود، لازم است کاربر از قبل وارد سیستم شده باشد.

در انتها توجه داشته باشید که با اجرای هر دستور ممکن است خطاهایی رخ دهند که در آن صورت می‌بایست پیام خطای مناسب چاپ شود. تعدادی از خطاهای مرسوم برنامه به شرح زیر هستند:

- عدم وجود مسیر عنصر مشخص شده در دستور
 - عدم وجود مجوز دسترسی خواندن/نوشتن به عنصر مشخص شده برای کاربر (با توجه به نوع دستور)
 - عدم وجود کاربری که در هنگام اجرای یک دستور، انتظار می‌رود وجود داشته باشد
 - تکراری بودن نام کاربر در هنگام ثبت نام و تکراری بودن نام گروه یا نام عنصر در هنگام ایجاد آن‌ها
 - صحیح نبودن نام کاربری یا گذرواژه
 - عدم وجود سطح دسترسی مناسب برای کاربر در هنگام اجرای دستوراتی که نیاز به سطح دسترسی خاصی دارند
- توجه داشته باشید که خطاهای بالا بخشی از خطاهای ممکن هستند و در صورت وجود خطایی غیر از خطاهای بالا، شما می‌بایست آن را در نظر بگیرید.

دستورات مربوط به مدیریت کاربران (۲۰٪ از نمره‌ی کل)

• اضافه کردن کاربر

برای اضافه کردن هر کاربر به سیستم، باید دستور زیر وارد شود:

```
add_user {username} {password}
```

نام کاربری در سیستم یکتاست. همچنین این برای این دستور به سطح دسترسی مدیر سیستم یا کاربر ممتاز نیاز است.

• ورود

هر کاربر می‌تواند با وارد کردن دستور زیر، وارد حساب کاربری خود شود:

```
login {username} {password}
```

توجه کنید که برای ورود یک کاربر، می‌بایست کاربر دیگری در سیستم وارد نباشد.

• خروج

هر کاربر می‌تواند با وارد کردن دستور زیر، از سیستم خارج شود:

```
logout
```

● اضافه کردن گروه

برای ساختن گروه جدید باید دستور زیر وارد شود :

```
add_group {group_name}
```

نام گروه در سیستم یکتاست. همچنین به صورت پیش فرض، کاربر سازنده گروه، عضو این گروه می باشد.

● اضافه کردن کاربر به گروه

با استفاده از دستور زیر، می توان یک کاربر را به یک گروه اضافه کرد:

```
add_to_group {group_name} {username}
```

توجه کنید کاربری که این دستور را اجرا می کنید باید عضو گروه مشخص شده باشد.

دستورات مربوط به مدیریت فایل (۴۰٪ دستورات فایل و ۳۰٪ دستورات مجوزهای دسترسی)

برنامه ی شما از یک پوشه برای بارگذاری و بارگیری فایل های واقعی استفاده می کند که مسیر این پوشه در زمان اجرا و به عنوان آرگومان^{۱۹} اول به برنامه ی شما داده می شود. هنگام بارگذاری نام فایل باید نام یکی از فایل های موجود در این پوشه باشد. هنگام بارگیری باید یک فایل با نام مشخص شده در این پوشه ایجاد شود.

● بارگذاری فایل

با اجرای دستور زیر، کاربر می تواند فایل مورد نظر خود را در مسیری معین، بارگذاری کند:

```
upload {real_file_name} {path}
```

● بارگیری فایل

با اجرای دستور زیر، کاربر می تواند فایل مورد نظر خود را از مسیری معین، بارگیری کند:

```
download {path} {output_name}
```

● ایجاد پوشه

هر کاربر می تواند با اجرای دستور زیر، پوشه ای را در مسیر فعلی به وجود بیاورد:

```
make_directory {directory_name}
```

¹⁹ Argument

● نسخه‌برداری از یک عنصر

هر کاربر با اجرای دستور زیر، می‌تواند از یک فایل نسخه‌برداری کند:

```
copy {source_path} {destination_path}
```

توجه کنید که این دستور نیاز به به دسترسی خوانند از مسیر مبدأ و نوشتن در مسیرهای مقصد دارد.

● جابجا کردن یک عنصر

کاربر می‌تواند با استفاده از دستور زیر، فایل موجود در مسیر مبدأ را به مسیر مقصد، جابجا کند.

```
move {source_path} {destination_path}
```

توجه کنید که جابه‌جایی فایل نیاز به دسترسی نوشتن در مسیرهای مبدأ و مقصد دارد.

● حذف کردن یک عنصر

کاربر می‌تواند با استفاده از دستور زیر، فایل موجود در مسیری معین را حذف کند. توجه کنید که مسیر مشخص شده می‌تواند مسیر یک file یا پوشه باشد.

```
remove {path}
```

● چاپ کردن پوشه فعلی

کاربر می‌تواند با استفاده از دستور زیر، مسیر پوشه‌ای که در حال حاضر در آن وجود دارد را مشاهده کند:

```
print_directory
```

● تغییر دادن پوشه فعلی

کاربر می‌تواند با استفاده از دستور زیر، پوشه‌ی فعلی خود را تغییر دهد.

```
change_directory {path}
```

● مشاهده جزئیات یک یا چند عنصر

کاربر می‌تواند با استفاده از دستور زیر، جزئیات یک یا چند عنصر را مشاهده کند.

```
elem_info [path]
```

در صورتی که این دستور بدون آرگومان استفاده شود، باید لیستی از همه‌ی عناصر موجود در آن مسیر با فرمت زیر چاپ شود. در صورت وجود آرگومان، کافی است تنها جزئیات آن عنصر خاص با فرمت زیر چاپ شود.

```
{dir | file} {owner_permissions} {group_permissions} {owner_name} {group_name}
{file_name}
{(owner|group)_permissions} = {{read | -},{write | -}}
```

برای مثال داریم:

```
dir read,write read,- amir AP /amir/bin
```

عبارت فوق بدان معنا می باشد که در پوشه ای که این دستور اجرا شده است، یک پوشه، به نام /amir/bin وجود دارد که کاربر مالک آن amir بوده و گروه مالک AP است. مجوز read و write برای کاربر مالک و دسترسی read برای گروه مالک تعیین شده است (کاراکتر - به معنای عدم وجود مجوز write برای گروه مالک این عنصر است).

• نمایش حجم فایل

با استفاده از این دستور می توان حجم فایل مشخص شده را به Kb مشاهده کرد.

```
file_size {path}
```

• تغییر حالت یک عنصر

کاربر می تواند با اجرای دستور زیر، مجوزهای دسترسی به یک عنصر را با استفاده از گزینه های مناسب، تغییر دهد.

```
change_mode {group | user} {+ | -} {read | write | rw} {file | directory_path}
```

با استفاده از این دستور می توان برای کاربر مالک یا گروه مالک عنصر مشخص شده یک دسترسی خاص را به آن اضافه یا از آن حذف کرد. آرگومان اول دستور، مشخص کننده ی گروه یا کاربر است. آرگومان سوم یکی از علامت های - یا + می باشد. علامت + به معنی اضافه کردن مجوز و علامت - به معنای حذف مجوز است. آرگومان چهارم، نوع مجوز مورد نظر است. در انتها، عنصر مورد نظر ذکر می شود. به عنوان مثال داریم:

```
change_mode group + write /userName/example.txt
```

با اجرای این دستور، به گروهی که مالک فایل /userName/example.txt است، مجوز دسترسی نوشتن بر روی این فایل داده می شود. دقت شود که تنها مالک فایل می تواند حالت فایل را تغییر دهد.

• تغییر مالک یک عنصر

مالک یک عنصر، می تواند با اجرای دستور زیر، مالکیت این پوشه را به کاربر دیگری واگذار کند. پس از این تغییر مالک قدیمی دیگر به عنصر دسترسی ندارد.

```
change_owner {username} {file/directory_path}
```

● تغییر گروه یک عنصر

مالک یک فایل، می‌تواند با اجرای دستور زیر، گروه مالک عنصر را که مجوز دسترسی به آن دارند را تغییر دهد. تمام مجوزها از گروه قدیمی سلب می‌شود. توجه کنید که لزومی وجود ندارد که مالک عنصر عضوی از گروه باشد.

```
change_group {group_name} {file/directory_path}
```

عملیات‌های مربوط به کاربر ممتاز و مدیر سیستم (۱۰٪ کل)

● ترفیع²⁰ یک کاربر

نقش یک کاربر مشخص، با اجرای دستور زیر، می‌تواند ترفیع یابد. تنها کاربر ممتاز است که مجوز تغییر حالت کاربری به مدیر و کاربر ممتاز را دارد. دقت شود که در این دستور اجازه تنزل رتبه وجود ندارد. برای مثال، اگر کاربر amir یک superuser باشد، امکان اجرای دستور promote amir admin وجود ندارد.

```
promote {username} {superuser | admin}
```

● تنزل²¹ یک کاربر

نقش یک کاربر مشخص، با اجرای دستور زیر، می‌تواند تنزل یابد. تنها کاربر ممتاز است که مجوز تغییر حالت کاربری به مدیر و کاربر عادی را دارد. دقت شود که در این دستور اجازه ترفیع رتبه وجود ندارد.

```
demote {username} {admin | normal}
```

● نمایش فهرست کاربران

کاربر ممتاز و مدیر سیستم می‌توانند با استفاده از دستور زیر، فهرستی از کاربران فعلی سیستم را مشاهده کنند.

```
userlist
```

● مشاهده حجم مصرفی کل سیستم

کاربر ممتاز و مدیر سیستم می‌توانند با اجرای دستور زیر، حجم مصرفی از حساب کل سیستم را مشاهده کنند.

```
storage
```

²⁰ Promote

²¹ Demote

نکات تکمیلی

- به ازای تمام دستورات ذکر شده، در صورت وارد کردن داده‌ی غیرمعتبر برنامه باید پیام خطای مناسبی را برای کاربر چاپ کند (مثلاً کم یا زیاد بودن تعداد آرگومان‌های داده شده به دستور). همچنین توجه داشته باشید که تعدادی از دستورات دسترسی‌های خاصی دارند یا نیازمند مسیر مشخصی هستند که این پارامترها می‌بایست بررسی شوند و در صورت خطا، پیام مناسب چاپ شود.
- به صورت پیش‌فرض یک کاربر ممتاز با نام کاربری root و گذرواژه‌ی root در سیستم وجود دارد. امکان حذف این کاربر از سیستم وجود ندارد. این کاربر عضو گروه root است و این گروه مجوز دسترسی خواندن و نوشتن به همه‌ی فایل‌های سیستم را دارد.
- توجه کنید که در فاز بعدی این پروژه از کدی که در این فاز زده‌اید استفاده می‌شود، بنابراین به کیفیت کد خود خصوصاً نگهداشت‌پذیری²² و بازبهره‌پذیری²³ آن دقت کنید.
- توجه کنید که در هنگام اجرای دستوراتی که با فایل یا پوشه کار می‌کنند، اگر مسیر مشخص شده یک پوشه باشد، دستور اجرا شده باید بر روی تمامی فایل‌ها و پوشه‌هایی که درون پوشه‌ی مشخص شده قرار دارند اجرا شود. مثلاً با اجرای دستور remove dir1 اگر dir1 مسیر یک پوشه باشد، علاوه بر این پوشه دستور remove باید بر روی تمامی فایل‌ها و پوشه‌های درون این پوشه هم اجرا شوند. بنابراین پیش از اجرای دستور باید وجود مجوز دسترسی نوشتن به تمامی آن‌ها برای کاربر نیز بررسی شود.
- ترتیب چاپ در دستوراتی که لیست چاپ می‌کنند (userlist و elem_info) اهمیت ندارد.
- دسترسی یک کاربر به یک عنصر خاص اجتماع‌ای از دسترسی‌های اوست. برای مثال، اگر کاربر amir عضو گروه AP باشد و گروه AP به فایل /amir/a.txt دسترسی read داشته باشد. همچنین او کاربر مالک فایل amir/a.txt باشد و برای کاربر مالک آن file تنها دسترسی write وجود داشته باشد، کاربر amir به این فایل دسترسی read و write دارد.

شیوه‌ی تحویل

- برنامه‌ی شما باید در سیستم عامل لینوکس، با کامپایلر ++g و با استاندارد ++11 c ترجمه و اجرا شود.
- فایل‌های برنامه‌ی خود را با نام A4-SID.zip در صفحه‌ی CECM درس بارگذاری کنید که در آن SID شماره دانشجویی شماست. لطفاً از روش‌های دیگر فشرده‌سازی مانند rar یا tar.gz استفاده نکنید.
- این پروژه باید حتماً به روش شی‌گرایی و به صورت Multi File باشد و استفاده از makefile اجباری می‌باشد.
- به فرمت و نام فایل‌های خود دقت کنید.
- سوالات خود را در فروم درس مطرح کنید تا دوستانتان نیز از آن‌ها استفاده کنند.
- از صحت فرمت ورودی‌ها و خروجی‌های برنامه‌ی خود مطمئن شوید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.

²² Maintainability

²³ Reusability